Short Paper: Privacy-preserving ECC-based grouping proofs for RFID*

Lejla Batina[1,3], Yong Ki Lee[2], Stefaan Seys[3], Dave Singelée[3], and Ingrid Verbauwhede[3]

[1] CS Department/Digital Security group
Radboud University Nijmegen, The Netherlands
[2] Samsung Electronics Research and Development, South-Korea
[3] Department of Electrical Engineering/SCD-COSIC & IBBT
University of Leuven, Belgium
email: firstname.lastname@esat.kuleuven.be

Abstract. The concept of grouping proofs has been introduced by Juels to permit RFID tags to generate a verifiable proof that they have been scanned simultaneously, even when readers or tags are potentially untrusted. In this paper, we extend this concept and propose a narrow-strong privacy-preserving RFID grouping proof and demonstrate that it can easily be extended to use cases with more than two tags, without any additional cost for an RFID tag. Our protocols rely exclusively on the use of Elliptic Curve Cryptography (ECC). To illustrate the implementation feasibility of our proposed solutions, we present a novel ECC hardware architecture designed for RFID.

Keywords: RFID, Authentication, Grouping Proofs, ECC, Privacy

1 Introduction

The concept of RFID grouping proofs, also denoted by yoking proofs, was introduced by Juels [1]. The motivation comes from any application that requires the proof that two or more entities are present. For example, there could be a legal requirement that certain medication should be distributed together with a brochure describing its side-effects. A technical solution to this problem is to attach RFID tags to both the medication and the brochures, and create grouping proofs when they are scanned simultaneously. The pharmacist then stores these grouping proofs as evidence, to transmit them to the government for verification. Other use cases include monitoring of groups of hardware components that needs to be shipped together, coupling a physical person via his passport

* This work was supported in part by the IAP Programme P6/26 BCRYPT of the Belgian State, by FWO project G.0300.07, by the European Commission under contract number ICT-2007-216676 ECRYPT NoE phase II, and by the K.U. Leuven-BOF (OT/06/40).

to his boarding pass, or – in the military context – only enabling weaponry or equipment when an appropriate group of entities is present.

Recently, other work proposed in the literature improved the computational complexity of Juels' protocol, and considered other requirements such as privacy and forward security. The common property for all the schemes proposed so far is the use of symmetric-key primitives. Such schemes are however often not scalable, and entail several security (*e.g.*, cloning attacks) and/or privacy problems (*e.g.*, it is proven that one needs public-key cryptography to achieve a certain level of privacy protection [6]). In contrast to this, the privacy-preserving grouping-proof protocols we propose in this paper rely exclusively on the use of public-key cryptography. More in particular, they are founded on the ECC-based ID-transfer protocol proposed by Lee *et al.* [2], as this scheme entails interesting security and privacy properties.

This paper is organized as follows. In Sect. 2 we describe our assumptions and adversary model. Our grouping-proof protocol is given in Sect. 3. A novel architecture for an ECC processor suitable for RFID is outlined in Sect. 4. We conclude our work in Sect. 5.

2 Assumptions and adversary model

In our setting, there are three distinct parties involved: the set of tags, the reader, and a trusted verifier. The former two will engage in a protocol run, which results in the construction of the grouping proof. This proof is then verifiable (offline) by the trusted verifier.

Due to the "simultaneously scanned" requirement, the notion of time is very important as already pointed out by Juels [1]. We assume that both the reader and the tags measure the round-trip-time during the execution of the protocol. If this round-trip-time exceeds a particular threshold, the protocol is aborted and the proof remains incomplete. Note that due to these timeouts, the protocol will always terminate.

We assume that the verifier is trusted and the public-key Y of the verifier is a publicly known system parameter. Only the verifier knows the corresponding private-key y. Knowledge of y is a necessary requirement to check the correctness of a grouping proof. The result of a verification claim is failure, or it reveals the identities of the involved tags. In this case the verifier stores and timestamps the grouping proof (enabling temporal ordering of the proofs). The task of the reader is to coordinate the execution of the protocol, collect the grouping proof and forward it to the verifier. The reader is not necessarily trusted by the tags or the verifier.

It should be impossible to generate a valid grouping proof without the involved tags actually participating in the protocol. Without loss of generality, we assume that there are only two participating tags. To avoid impersonation attacks or fake grouping proofs, one needs to prevent the following potential attack scenarios:

Compromised tag: One tag is compromised, the reader is non-compromised.

Man-in-the-middle attack: The reader is compromised (the tags are honest).
Colluding reader and tag: The reader and one of the tags are compromised.
Colluding tags: The reader is non-compromised, both tags are compromised. The tags can exchange some messages in advance (*e.g.*, via another reader), but do not know each other's private key.
Replay attack performed by an outsider: An eavesdropper scans two non-compromised tags simultaneously and replays the copied message-flow to impersonate the two tags.

Note that if all tags and the reader are compromised, this enables the adversary to generate valid grouping proofs without simultaneously scanning the tags. We also do not consider the attack where an adversarial reader scans two non-compromised tags, and forwards the grouping proof at a later time to the verifier (*i.e.* to have an incorrect timestamp being added to the grouping proof). Note that the grouping proofs that are proposed in this paper, do not prove that the tags are located in physical proximity to one another. An adversary can use multiple readers, and forward messages between these devices, to simultaneously scan tags at remote locations. Besides the large effort and cost, the effect of this attack is limited due to the timeout mechanism.

In the design of our protocol, we also want to achieve *untraceability*, in which the (in)equality of two tags must be impossible to determine. Only the trusted verifier should be able to check a grouping proof. To evaluate the privacy of our scheme, we adopt the adversarial capabilities from the framework of Vaudenay [6].

3 ECC-based grouping-proof protocol with colluding tag prevention

3.1 Notation

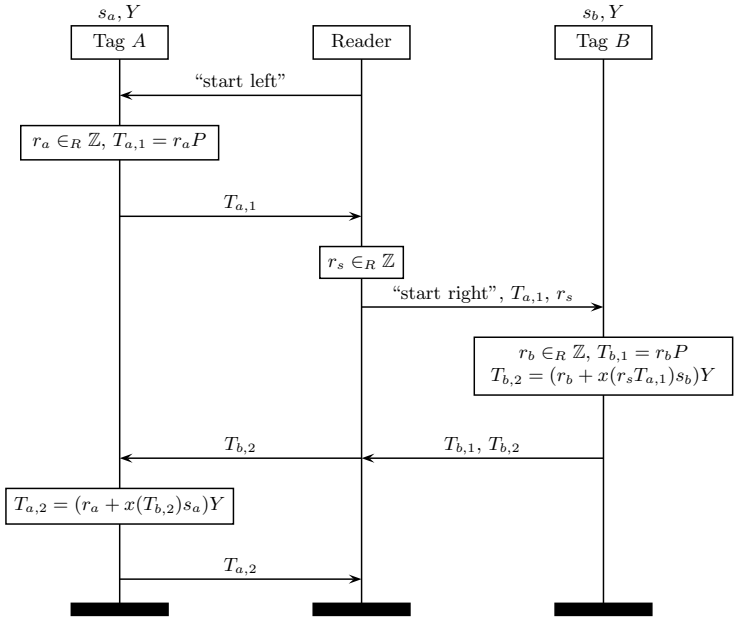
Let us first introduce the notation used in this work. We denote P as the base point on a Elliptic Curve, and y and $Y(= yP)$ are the trusted verifier's private-key and public-key pair, where yP denotes the point derived by the point multiplication operation on the Elliptic Curve group. We use the notation $x(T)$ to denote the x-coordinate of the point T on the elliptic curve, and $\dot{r}_s$ to denote the non-linear mapping $x(r_sP)$, with P the base point of the elliptic curve. The values s_t and $S_t(= s_tP)$ are tag t's private-key and public-key.

3.2 Protocol description

In this section, we propose a privacy-preserving ECC-based grouping-proof protocol with colluding tag prevention (denoted by CTP). It allows a pair of RFID tags (denoted by tag A and B) to prove that they have been scanned simultaneously.

The two-party CTP protocol is shown in Fig. 1. During the entire execution of the protocol, the tags and/or the reader abort when a timeout occurs, or

Fig. 1. Two-party grouping-proof protocol with colluding tag prevention (CTP).



when they receive the EC point at infinity. The protocol works as follows. The reader first sends the messages *"start left"* and *"start right"* to indicate the role of the tags in the protocol. Next, tag A generates a random number r_a and the corresponding EC point $T_{a,1}$. Tag B carries out similar operations. Both tags also compute a response. The response $T_{b,2}$ depends on the private-key s_b, the random number r_b, the x-coordinate of the challenge $T_{a,1}$, and a random challenge r_s generated by the reader. The response $T_{a,2}$ depends on the private-key s_a, the random number r_a, and the x-coordinate of the challenge $T_{b,2}$. The grouping proof, collected by the reader, consists of the tuple $(T_{a,1}, T_{a,2}, r_s, T_{b,1}, T_{b,2})$.

To verify the grouping proof constructed by tag A and B, the verifier first checks that the proof was not used before (to detect replay attacks) and then performs the following computations:$s_a P = (y^{-1} T_{a,2} - T_{a,1}) x(T_{b,2})^{-1}$ and $s_b P = (y^{-1} T_{b,2} - T_{b,1}) x(r_s T_{a,1})^{-1}$. If the public keys of A and B (S_a and S_b respectively) are registered in the database of the verifier, the grouping proof is accepted and a timestamp is added.

3.3 Extension to $n > 2$ parties

The two-party CTP grouping-proof protocol shown in Fig. 1 can be easily extended to multiple tags ($n > 2$). The output of each tag is then used as input for the "next" tag in the chain, as shown in Fig. 2. This procedure is repeated until all tags are scanned. The last tag in the chain (denoted by tag Z) sends $T_{z,2}$

to tag A, which then computes its response $T_{a,2}$. The grouping proof consists of the following tuple: $(T_{a,1}, T_{a,2}, \ldots, T_{i,1}, T_{i,2}, \ldots, T_{z,1}, T_{z,2})$. To check the correctness of the grouping proof, the verifier performs similar operations as with the two-party CTP grouping-proof protocol.

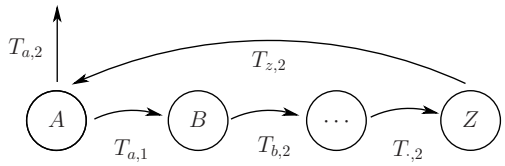


Fig. 2. Chain of grouping proofs.

3.4 Analysis

Due to its construction, our CTP grouping-proof protocol inherits the security properties of the ID-transfer protocol [2]. The latter is designed to provide secure entity authentication in the setting of an active adversary, and can be shown to be equivalent to the Schnorr protocol [5] regarding impersonation resistance. One can demonstrate that to impersonate a tag in either of our attack scenarios, the adversary needs to know the private-key of that particular tag (or be able to solve the Decisional Diffie-Hellman (DDH) problem).

The same argumentation as above can be used to demonstrate the privacy properties of the CTP grouping-proof protocol. Since the ID-transfer protocol offers privacy protection against a narrow-strong adversary, untraceability can even be guaranteed if the challenges of the ID-transfer protocol are controlled by the adversary. As a direct consequence, the CTP grouping-proof protocol is also narrow-strong privacy-preserving. [1]

In our protocol, each tag i has to perform two EC point multiplications to create the output $T_{i,1}$ and $T_{i,2}$. The workload of a tag is independent of the number of tags n involved in the protocol. Another interesting observation is that an n-party grouping proof exactly contains $2n$ EC points. The bitlength of the grouping proof is thus linearly dependent on the number of tags n. Note however that there is a practical upper limit on the number of tags n that can be scanned simultaneously. If n is very large, a timeout could occur in tag A before the protocol has terminated.

4 Implementation

In order to show the feasibility of the proposed protocols for RFID tags, we analyze a hardware implementation of our solutions. The EC processor we present

[1] More details can be found in an extended version of this paper, see https://www.cosic.esat.kuleuven.be/publications/.

in this paper has a novel architecture that features the most compact and at the same time the fastest solution when compared to previous work.

The overall architecture is shown in Fig. 4. The processor consists of a micro controller, a bus manager and an EC processor (ECP). It is connected with a front-end module, a random number generater (RNG), ROM and RAM. The ROM stores program codes and data that may include a tag's private key, the server's public key and system parameters. The program is basically a grouping proof for a tag or an authentication protocol.

The architecture of MALU with the required registers is shown in Fig. 3. Here the registers in the MALU are combined with the external ones to reduce the total number of registers.

The new ECP architecture is similar to the one presented in [3]. Further optimizations are performed in the register file and the Modular ALU (MALU). The EC processor presented in [3] uses a MALU which performs modular addition and multiplications, and it reuses the logic of modular multiplications for modular squaring operations. On the other hand, the new MALU presented here includes a specialized squarer logic. Since the modular squaring can be completed in one cycle on a dedicated squarer, the performance can be substantially increased with an overhead of the square logic. Moreover, in the new architecture the size of register file is reduced to 5×163 bits from 6×163 bits as we are using ECC over $GF(2^{163})$. In addition, the cost for the merged squarer is 558 gates only.

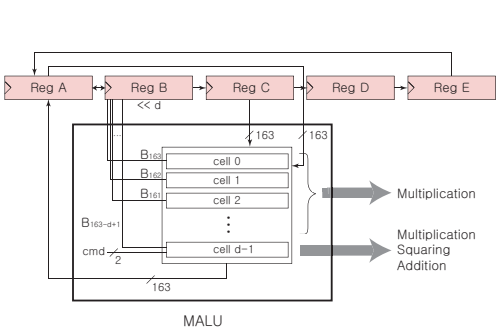


Fig. 3. MALU architecture with register file.

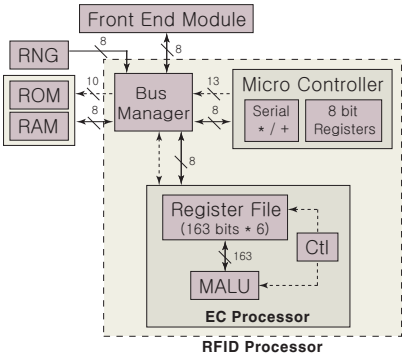


Fig. 4. RFID Processor Architecture.

To give an idea about the improvements for the new MALU we compared three different versions of MALU in area. For MALU without and with a squarer the gate counts are 913 and 1 636 gates respectively (digit size $d = 1$). The latter can be improved to 1 471 when squarer and multiplier are sharing the XOR array. Adding a squarer results in a small increase in area (for MALU) but total area is reduced due to the reduction in the number of registers.

The performance comparison is also made with the work in [3] for the digit size of 4 in the MALU for both architectures. This work achieves about 24% better performance with a smaller circuit area, and the energy consumption is much smaller. In particular, the size of our ECP processor is estimated to 14,566 kgates. We used a $0.13\mu m$ CMOS technology, and the gate area does not include RNG, ROM and RAM which are required to store or run programmed protocols. The area specifies a complete EC processor with required registers. The required number of cycles for scalar multiplication is 78 544. Assuming an operating frequency of $700KHz$ expected power consumption is around $11.33\mu W$ per point multiplication. The performance result for our protocol is estimated to 295 ms.

5 Conclusions

We presented an efficient privacy-preserving grouping-proof protocol for RFID based solely on ECC. The protocol enables two tags to generate a proof that both were scanned (virtually) simultaneously. The only complex operations required from the tags are the generation of a random number and EC point multiplications. We also show how to extend the protocol to multiple ($n > 2$) tags. In addition, we presented a hardware architecture that demonstrates the feasibility of the protocols even for a passive tag.

References

1. A. Juels. "Yoking-Proofs" for RFID Tags. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW '04)*, pages 138–143. IEEE Computer Society, 2004.
2. Y. K. Lee, L. Batina, D. Singelée, and I. Verbauwhede. Low-Cost Untraceable Authentication Protocols for RFID (extended version). In S. Wetzel, C. N. Rotaru, and F. Stajano, editors, *Proceedings of the 3rd ACM Conference on Wireless Network Security (WiSec '10)*, pages 55–64. ACM, 2010.
3. Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede. Elliptic Curve Based Security Processor for RFID. *IEEE Transactions on Computer*, 57(11):1514–1527, November 2008.
4. D. Naccache, N. P. Smart, and J. Stern. Projective coordinates leak. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology (EUROCRYPT '04)*, volume 3027 of *Lecture Notes in Computer Science*, pages 257–267. Springer, 2004.
5. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In G. Brassard, editor, *Advances in Cryptology (CRYPTO '89)*, Lecture Notes in Computer Science, LNCS 435, pages 239–252. Springer-Verlag, 1989.
6. S. Vaudenay. On privacy models for RFID. In *Advances in Cryptology (ASIACRYPT'07)*, Lecture Notes in Computer Science, LNCS 4833, pages 68–87. Springer-Verlag, 2007.